

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курский государственный университет»

Кафедра программного
обеспечения и администрирования
информационных систем

Направление подготовки
математическое обеспечение и
администрирование
информационных систем

Форма обучения очная

Отчет
по лабораторной работе №1
«Линейное исполнение программ. Арифметические и поразрядные
логические операции над байтами»

Выполнил:

студент группы 213

Водолад Д.В.

Проверил:

доцент кафедры ПОиАИС

Кривонос А.В.

Курск, 2021

Цели работы:

1. изучение принципов функционирования памяти и микропроцессора компьютера при последовательном исполнении команд программы;
2. приобретение навыков использования арифметических команд при написании ассемблерных программ;
3. приобретение навыков использования поразрядных логических команд при написании ассемблерных программ;
4. получение представления об особенностях обработки данных разных размерностей и режимах доступа к данным при выполнении арифметических операций.

Задание:

- 1) Написать программу на языке ассемблера, которая выполняет арифметические и поразрядные логические операции над байтовыми значениями.
 - 1.1) В сегменте данных определить два байтовых значения 10 и 27. В десятичной системе счисления и однобайтовое произвольное число в двоичной системе счисления.
 - 1.2) В сегменте данных зарезервировать байтовые ячейки для хранения суммы и разности с нулевыми первоначальными значениями, двухбайтовую ячейку для хранения произведения с единичным первоначальным значением, две байтовые ячейки для хранения остатка от деления и частного с произвольными первоначальными значениями.
 - 1.3) Выполнить сложение 10 и 27; полученный результат записать в соответствующую ячейку памяти.
 - 1.4) Выполнить вычитание 10 и 27; полученный результат переслать в соответствующую ячейку памяти.
 - 1.5) Изменить знак второго числа (27) и снова выполнить операцию вычитания 10 и -27.
 - 1.6) Выполнить умножение 10 и -27 с учетом знака; результат записать в соответствующую ячейку памяти. Выполнить умножение 10 и -27 без учета знака.

- 1.7) Выполнить деление 27 на 10; полученные результаты записать в соответствующие ячейки памяти.
- 1.8) Переписать его в регистр, установить 2 любых бита в единицу, инвертировать все, сбросить 3 любых бита.
- 1.9) Полученный результат продублировать в другом регистре, сложить получившиеся значения по модулю два.
- 2) На основе исходной программы получить исполняемый файл. Выполнить программу по шагам с помощью отладчика TURBO DEBUGGER, описать изменение состояния регистров и ячеек памяти при выполнении программы. Обратить особое внимание на следующие моменты:
- 2.1) Как представляется число 27 и -27 в 16-ричной системе счисления?
- 2.2) Какое значение разности при выполнении вычитания в пунктах 1.4 и 1.5 задания?
- 2.3) Где размещается результат умножения 10 и -27?
- 2.4) В чем заключается разность произведения 10 и -27 при умножении со знаком и без учета знака?
- 2.5) В каких регистрах размещаются результаты деления 27 и 10, и чему равны значения частного и остатка от деления?
- 2.6) Чему равна маска для установки двух битов в единицу и почему?
- 2.7) Чему равна маска для сброса трех битов в ноль?

Текст программы

d_s segment

data1 db 10, 27

data2 db 0101B

el db 3

sum db 0

raz db 0

pr dw 1

mod1 db 0

divv db 0

d_s ends

c_s segment

assume ds:d_s,cs:c_s

begin:

mov ax,d_s

mov ds,ax

mov dl,data1 ;10+27

add dl,data1+1

mov sum,dl

mov al, data1; 10 - 27 signed

sub al, data1+1

mov raz, al

neg data1+1; 10 - (-27)

mov al, data1

sub al, data1+1

mov raz, al

mov al, data1 ; 10 * (-27)

imul data1+1

mov pr, ax

neg data1+1 ; 10/27

mov al, data1

mov ah, 0h

div data1+1

mov mod1, ah

mov divv, al

```
mov el, 3
```

```
or al, el
```

```
not al
```

```
mov el, 7
```

```
not el
```

```
and al, el
```

```
mov dl, al
```

```
xor dl, al
```

```
mov ah,4ch
```

```
int 21h
```

```
c_s ends
```

```
end begin
```

Вывод:

- 1) я изучил принцип функционирования памяти и микропроцессора компьютера при последовательном исполнении команд программы;
- 2) приобрёл навык использования арифметических команд при написании ассемблерных программ;
- 3) приобрёл навык использования поразрядных логических команд при написании ассемблерных программ;
- 4) получил представление об особенностях обработки данных разных размерностей и режимах доступа к данным при выполнении арифметических операций.